

MMM		MMM	TTTTTTTTTTTTTTTT	AAAAAAAAAA		AAAAAAAAAA		CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM		MMM	TTTTTTTTTTTTTTTT	AAAAAAAAAA		AAAAAAAAAA		CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM		MMM	TTTTTTTTTTTTTTTT	AAAAAAAAAA		AAAAAAAAAA		CCCCCCCCCCCC	PPPPPPPPPPPP	
MMMMMM	MMMMMM		TTT	AAA	AAA	AAA	AAA	CCC	PPP	PPP
MMMMMM	MMMMMM		TTT	AAA	AAA	AAA	AAA	CCC	PPP	PPP
MMMMMM	MMMMMM		TTT	AAA	AAA	AAA	AAA	CCC	PPP	PPP
MMM	MMM	MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	PPP
MMM	MMM	MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	PPP
MMM	MMM	MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	PPP
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM		MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM		MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM		MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCCCCCCCCCCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCCCCCCCCCCC	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCCCCCCCCCCC	PPP	

```
LL      000000  GGGGGGGG  IIIIII  000000
LL      000000
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LL      00      00  GG      II      00      00
LLLLLLLLLLLL  000000  GGGGGG  IIIIII  000000
LLLLLLLLLLLL  000000  GGGGGG  IIIIII  000000
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIII  SSSSSSSS
```


C 5
16-Sep-1984 02:23:24
14-Sep-1984 12:46:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]LOGIO.B32;1 Page 1 (1)

LOG
V04

```
0001 0
0002 0 MODULE LOGIO (LANGUAGE (BLISS32),
0003 0 IDENT = 'V04-000',
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1
0032 1 FACILITY: MTAACP
0033 1
0034 1 ABSTRACT:
0035 1 This module handles logical IO.
0036 1
0037 1
0038 1 ENVIRONMENT:
0039 1
0040 1 Starlet operating system, including privileged system services
0041 1 and internal exec routines.
0042 1
0043 1 --
0044 1
0045 1
0046 1
0047 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 14-JUL-1977
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1 V03-009 HH0041 Hai Huang 24-Jul-1984
0052 1 Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
0053 1
0054 1 V03-008 ROW0258 Ralph O. Weber 21-NOV-1983
0055 1 The Paul Painter Memorial Enhancement
0056 1 Named for one of the unfortunate customers who suffered much
0057 1 to determine the great UCBSL_MT_RECORD secret while trying to
```

```
58      0058 1  create a user-written magtape driver, this change eliminates
59      0059 1  use of the device dependent field, UCB$MT_RECORD in favor of
60      0060 1  the device independent field, UCB$L_RECORD.
61      0061 1
62      0062 1  V03-007 STJ3101 Steven T. Jeffreys, 24-May-1983
63      0063 1  Removed reference to obsolete IOSM_INTSKIP.
64      0064 1
65      0065 1  V03-006 MMD0172 Meg Dumont, 9-May-1983 15:14
66      0066 1  Fix to make USER_STATUS defined consistently within module
67      0067 1
68      0068 1  V03-005 MMD0101 Meg Dumont, 17-Feb-1983 12:56
69      0069 1  Use routine GET_DEV_NAME to get tape unit device name.
70      0070 1
71      0071 1  V03-004 MMD0004 Meg Dumont, 21-Jan-1983 12:32
72      0072 1
73      0073 1  Change COMPLETE_VIO so it always puts an ABORT status
74      0074 1  in the IO to complete
75      0075 1
76      0076 1  V03-003 MMD0003 Meg Dumont, 3-Jan-1983 15:39
77      0077 1  Add the modifier IOSM_CLRSEREXCP to all QIO's issued by MTAACP,
78      0078 1  necessary for the MSCP tape drives.
79      0079 1
80      0080 1  V03-002 MMD0002 Meg Dumont, 5-Nov-1982 16:38
81      0081 1  Support for read record reverse. Support for the streaming tape
82      0082 1  drives, which forces all outstanding I/O's to complete
83      0083 1  before processing continues on a Serious Exception. Support
84      0084 1  for completing all I/O's to the user when USER EOT mode is set.
85      0085 1
86      0086 1  V03-001 MMD0001 Meg Dumont, 2-Jul-1982 12:08
87      0087 1  Add RETURN to ISSUE_IO to fix getting INFO message
88      0088 1
89      0089 1  V02-004 REFORMAT Maria del C. Nasr 30-Jun-1980
90      0090 1
91      0091 1  00003 MCN0009 Maria del C. Nasr 20-Nov-1979 12:00
92      0092 1  The STATUS code from a QIO is returned in low 16 bits of
93      0093 1  longword. Fix check for STATUS to <0,16>.
94      0094 1
95      0095 1  !!
96      0096 1  !**
97      0097 1
98      0098 1  LIBRARY 'SYS$LIBRARY:LIB.L32';
99      0099 1
100     0100 1
101     0101 1  REQUIRE 'SRC$:MTADEF.B32';
102     0485 1
103     0486 1  FORWARD ROUTINE
104     0487 1  ADJTM : COMMON CALL NOVALUE, ! adjust tape mark count
105     0488 1  CHCK_IO_CLR_EXCP : COMMON_CALL NOVALUE, ! Check to get all io's from
106     0489 1  ! device and clear serious expt
107     0490 1  COMPLETE_VIO : COMMON_CALL NOVALUE, ! complete virtual io in error
108     0491 1  ISSUE_IO : LS$ISSUE_IO, ! issue IO
109     0492 1  READ_BLOCK : COMMON_CALL, ! read logical block
110     0493 1  READ_BLOCK_REVERSE : COMMON_CALL, ! read backwards one logical block
111     0494 1  REPOSITION : LS$REPOSITION NOVALUE, ! reposition tape
112     0495 1  RESTORE_POS : NOVALUE COMMON_CALL,
113     0496 1  SPACE : COMMON_CALL, ! space blocks
114     0497 1  SPACE_TM : COMMON_CALL NOVALUE, ! space tape marks
```



```
: 115      0498 1      UNBLOCK SPACE      : COMMON CALL NOVALUE,      ! unblock for SPACE_TM
: 116      0499 1      WRITE_BLOCK : COMMON CALL NOVALUE,      ! write logical block
: 117      0500 1      WRITE_TM      : NOVALUE L$WRITE_TM;      ! write one tape mark
: 118      0501 1
: 119      0502 1      EXTERNAL
: 120      0503 1      CURRENT_UCB : REF BBLOCK,      ! address of current unit control block
: 121      0504 1      IO_CHANNEL,      ! address of IO channel
: 122      0505 1      IO_STATUS,      ! IO status
: 123      0506 1      USER_STATUS : VECTOR [2];      ! user status
: 124      0507 1
: 125      0508 1      EXTERNAL ROUTINE
: 126      0509 1      GET_DEV_NAME      : COMMON_CALL NOVALUE,      ! given UCB addr get dev name
: 127      0510 1      IO_DONE,      ! complete IO
: 128      0511 1      MOUNT_VOL      : COMMON_CALL,      ! mount volume
: 129      0512 1      PRINT_OPR_MSG      : L$PRINT_OPR_MSG,      ! print an operator message
: 130      0513 1      RESET_UNIT      : COMMON_CALL,
: 131      0514 1      SYSSQIOW      : ADDRESSING_MODE (ABSOLUTE);      ! queue io request
: 132      0515 1
```

```
134 0516 1 GLOBAL ROUTINE READ_BLOCK (ADDR, LEN) : COMMON_CALL =
135 0517 1
136 0518 1 !++
137 0519 1
138 0520 1 FUNCTIONAL DESCRIPTION:
139 0521 1 This routine reads a logical record from magnetic tape.
140 0522 1
141 0523 1 CALLING SEQUENCE:
142 0524 1 READ_BLOCK(ARG1,ARG2)
143 0525 1
144 0526 1 INPUT PARAMETERS:
145 0527 1 ARG1 - address for data
146 0528 1 ARG2 - length to read
147 0529 1
148 0530 1 IMPLICIT INPUTS:
149 0531 1 IO_CHANNEL
150 0532 1
151 0533 1 OUTPUT PARAMETERS:
152 0534 1 ARG1 - address for data
153 0535 1
154 0536 1 IMPLICIT OUTPUTS:
155 0537 1 USER_STATUS, IO_STATUS
156 0538 1
157 0539 1 ROUTINE VALUE:
158 0540 1 0 - tm encountered
159 0541 1 1 - successful read
160 0542 1
161 0543 1 SIDE EFFECTS:
162 0544 1 none
163 0545 1
164 0546 1 ERRORS:
165 0547 1 Primary status is I/O error returned from driver
166 0548 1 SS$_FCPREADERR - read failure
167 0549 1
168 0550 1 --
169 0551 1
170 0552 2 BEGIN
171 0553 2
172 0554 2 EXTERNAL REGISTER
173 0555 2 COMMON_REG;
174 0556 2
175 0557 2 LOCAL
176 0558 2 STATUS; ! IO status
177 0559 2
178 0560 2 STATUS = ISSUE_IO(IO$_READLBLK, .ADDR, .LEN);
179 0561 2
180 0562 2 IF .STATUS
181 0563 2 OR
182 0564 2 .STATUS<0,16> EQLU SS$_DATAOVERUN OR .STATUS<0,16> EQLU SS$_ENDOFFTAPE
183 0565 2 THEN
184 0566 2 RETURN 1;
185 0567 2
186 0568 2 IF .STATUS<0,16> NEQU SS$_ENDOFFILE
187 0569 2 THEN
188 0570 2 BEGIN
189 0571 2 USER_STATUS[0] = .STATUS;
190 0572 2 USER_STATUS[1] = SS$_FCPREADERR;
```



```
: 191      0573 3      ERR_EXIT();
: 192      0574 2      END;
: 193      0575 2
: 194      0576 2      KERNEL_CALL(ADJTM, 1);
: 195      0577 2      RETURN 0;
: 196      0578 2
: 197      0579 1      END;
```

```
! tm encountered
! end of routine
```

```
.TITLE LOGIO
.IDENT \V04-000\
```

```
.EXTRN CURRENT_UCB, IO_CHANNEL
.EXTRN IO_STATUS, USER_STATUS
.EXTRN GET_DEV_NAME, IO_DONE
.EXTRN MOUNT_VOL, PRINT_OPR_MSG
.EXTRN RESET_UNIT, SYSSQIOW
.EXTRN SYSSCMKRNL
```

```
.PSECT $CODE$,NOWRT,2
```

```
0000 00000
7E      04      AC 7D 00002
          21      DD 00006
          0000V 30 00008
          0C      C0 0000B
          5E      50 E8 0000E
0838    8F      50 B1 00011
          07      13 00016
0878    8F      50 B1 00018
          04      12 0001D
          50      01 D0 0001F 1$:
          04      04 00022
0870    8F      50 B1 00023 2$:
          0E      13 00028
0000G   CF      50 D0 0002A
0000G   CF      0888 8F 3C 0002F
          00      BF 00036
          01      DD 00038 3$:
          01      DD 0003A
          5E      DD 0003C
          0000V  CF  9F 0003E
00000000G 9F    04  FB 00042
          50      D4 00049
          04      04 0004B
```

```
.ENTRY READ_BLOCK, Save nothing
MOVQ    ADDR, -(SP)
PUSHL   #33
BSBW    ISSUE_IO
ADDL2   #12, SP
BLBS    STATUS, 1$
CMPW    STATUS, #2104
BEQL    1$
CMPW    STATUS, #2168
BNEQ    2$
MOVL    #1, R0
RET
CMPW    STATUS, #2160
BEQL    3$
MOVL    STATUS, USER_STATUS
MOVZWL  #2184, USER_STATUS+4
CHMU    #0
PUSHL   #1
PUSHL   #1
PUSHL   SP
PUSHAB  ADJTM
CALLS   #4, @#SYSSCMKRNL
CLRL    R0
RET
```

```
: 0516
: 0560
:
:
: 0562
: 0564
:
:
: 0566
:
: 0568
:
: 0571
: 0572
: 0573
: 0576
:
:
: 0577
: 0579
```

```
; Routine Size: 76 bytes, Routine Base: $CODE$ + 0000
```

```
199 0580 1 GLOBAL ROUTINE READ_BLOCK_REVERSE (ADDR, LEN) : COMMON_CALL =
200 0581 1
201 0582 1 ++
202 0583 1
203 0584 1 FUNCTIONAL DESCRIPTION:
204 0585 1 This routine reads in reverse a logical record from magnetic tape.
205 0586 1
206 0587 1 CALLING SEQUENCE:
207 0588 1 READ_BLOCK(ARG1,ARG2)
208 0589 1
209 0590 1 INPUT PARAMETERS:
210 0591 1 ARG1 - address for data
211 0592 1 ARG2 - length to read
212 0593 1
213 0594 1 IMPLICIT INPUTS:
214 0595 1 IO_CHANNEL
215 0596 1
216 0597 1 OUTPUT PARAMETERS:
217 0598 1 ARG1 - address for data
218 0599 1
219 0600 1 IMPLICIT OUTPUTS:
220 0601 1 USER_STATUS, IO_STATUS
221 0602 1
222 0603 1 ROUTINE VALUE:
223 0604 1 0 - tm encountered
224 0605 1 1 - successful read
225 0606 1
226 0607 1 SIDE EFFECTS:
227 0608 1 none
228 0609 1
229 0610 1 ERRORS:
230 0611 1 Primary status is I/O error returned from driver
231 0612 1 SS$_FCPREADERR - read failure
232 0613 1
233 0614 1 --
234 0615 1
235 0616 2 BEGIN
236 0617 2
237 0618 2 EXTERNAL REGISTER
238 0619 2 COMMON_REG;
239 0620 2
240 0621 2 LOCAL
241 0622 2 STATUS; ! IO status
242 0623 2
243 0624 2 STATUS = ISSUE_IO(IO$_READLBLK OR IO$_M_REVERSE, .ADDR, .LEN);
244 0625 2
245 0626 2 IF .STATUS
246 0627 2 OR
247 0628 2 .STATUS<0,16> EQLU SS$_DATAOVERUN OR .STATUS<0,16> EQLU SS$_ENDOFTAPE
248 0629 2 THEN
249 0630 2 RETURN 1;
250 0631 2
251 0632 2 IF .STATUS<0,16> NEQU SS$_ENDOFFILE
252 0633 2 THEN
253 0634 2 BEGIN
254 0635 2 USER_STATUS[0] = .STATUS;
255 0636 2 USER_STATUS[1] = SS$_FCPREADERR;
```



```
! tm encountered
! end of routine
```

: 263 0644 1

```

265 0645 1 GLOBAL ROUTINE WRITE_BLOCK (ADDR, LEN) : COMMON_CALL NOVALUE =
266 0646 1
267 0647 1 ++
268 0648 1
269 0649 1 FUNCTIONAL DESCRIPTION:
270 0650 1 This routine writes one logical block.
271 0651 1
272 0652 1 CALLING SEQUENCE:
273 0653 1 WRITE_BLOCK(ARG1,ARG2)
274 0654 1
275 0655 1 INPUT PARAMETERS:
276 0656 1 ARG1 - address of data block to write
277 0657 1 ARG2 - length of data block to write
278 0658 1
279 0659 1 IMPLICIT INPUTS:
280 0660 1 IO_CHANNEL
281 0661 1
282 0662 1 OUTPUT PARAMETERS:
283 0663 1 one block written
284 0664 1
285 0665 1 IMPLICIT OUTPUTS:
286 0666 1 IO_STATUS, USER_STATUS
287 0667 1
288 0668 1 ROUTINE VALUE:
289 0669 1 none
290 0670 1
291 0671 1 SIDE EFFECTS:
292 0672 1 SSS_FCPWRITERR - write failure
293 0673 1
294 0674 1 --
295 0675 1
296 0676 2 BEGIN
297 0677 2
298 0678 2 EXTERNAL REGISTER
299 0679 2 COMMON_REG;
300 0680 2
301 0681 2 LOCAL
302 0682 2 STATUS; ! IO status
303 0683 2
304 0684 2 STATUS = ISSUE_IO(IO$_WRITEBLK, .ADDR, .LEN);
305 0685 2
306 0686 2 IF NOT .STATUS AND .STATUS<0,16> NEQ SSS_ENDOFTAPE
307 0687 2 THEN
308 0688 3 BEGIN
309 0689 3 USER_STATUS[0] = .STATUS;
310 0690 3 USER_STATUS[1] = SSS_FCPWRITERR;
311 0691 3 ERR_EXIT();
312 0692 2 END;
313 0693 2
314 0694 1 END; ! end of routine
```

```

7E 04 0000 00000
AC 7D 00002
```

```

.ENTRY WRITE_BLOCK, Save nothing
MOVQ ADDR,--(SP)
```

```

: 0645
: 0684
```


LOG10
V04-000

K 5
16-Sep-1984 02:23:24
14-Sep-1984 12:46:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]LOG10.B32;1
Page 9
(4)

		20	DD	00006	PUSHL	#32	:
		0000V	30	00008	BSBW	ISSUE_10	:
	5E	0C	C0	0000B	ADDL2	#12, SP	:
0878	15	50	58	0000E	BLBS	STATUS, 1\$	0686
	8F	50	B1	00011	CMPW	STATUS, #2168	:
		0E	13	00016	BEQL	1\$:
0000G	CF	50	D0	00018	MOVL	STATUS, USER_STATUS	0689
0000G	CF	8F	3C	0001D	MOVZWL	#2208, USER_STATUS+4	0690
		00	BF	00024	CHMU	#0	0691
		04	00026	1\$:	RET		0694

; Routine Size: 39 bytes, Routine Base: \$CODE\$ + 009A

; 315 0695 1

LOG
V04

```
317 0696 1 GLOBAL ROUTINE SPACE (NUMBER) : COMMON_CALL =
318 0697 1
319 0698 1 ++
320 0699 1
321 0700 1 FUNCTIONAL DESCRIPTION:
322 0701 1 This routine spaces a given number of records in either direction.
323 0702 1
324 0703 1 CALLING SEQUENCE:
325 0704 1 SPACE(ARG1)
326 0705 1
327 0706 1 INPUT PARAMETERS:
328 0707 1 ARG1 - number of records to space
329 0708 1 ( positive means forward space, negative means backspace )
330 0709 1
331 0710 1 IMPLICIT INPUTS:
332 0711 1 IO_CHANNEL
333 0712 1
334 0713 1 OUTPUT PARAMETERS:
335 0714 1 none
336 0715 1
337 0716 1 IMPLICIT OUTPUTS:
338 0717 1 IO_STATUS, USER_STATUS
339 0718 1 Tape positioned accordingly
340 0719 1
341 0720 1 ROUTINE VALUE:
342 0721 1 0 - end of file
343 0722 1 1 - successful
344 0723 1
345 0724 1 SIDE EFFECTS:
346 0725 1 SS$_FCPSPACERR - space failure
347 0726 1
348 0727 1 --
349 0728 1
350 0729 2 BEGIN
351 0730 2
352 0731 2 EXTERNAL REGISTER
353 0732 2 COMMON_REG;
354 0733 2
355 0734 2 LOCAL
356 0735 2 TM, ! number of tape marks
357 0736 2 STATUS; ! io status
358 0737 2
359 0738 2 STATUS = ISSUE_IO(IO$_SKIPRECORD, .NUMBER, 0);
360 0739 2
361 0740 2 IF NOT .STATUS
362 0741 2 THEN
363 0742 3 BEGIN
364 0743 3
365 0744 3 IF .STATUS<0,16> EQL SS$_ENDOFFILE
366 0745 3 THEN
367 0746 4 BEGIN
368 0747 4 TM = 1; ! encountered one spacing forward
369 0748 4
370 0749 4 IF .NUMBER LSS 0
371 0750 4 THEN
372 0751 4 TM = -1; ! encountered one backspacing
373 0752 4
```


374	0753	4
375	0754	4
376	0755	4
377	0756	4
378	0757	4
379	0758	3
380	0759	3
381	0760	3
382	0761	3
383	0762	3
384	0763	3
385	0764	3
386	0765	3
387	0766	3
388	0767	2
389	0768	2
390	0769	2
391	0770	2
392	0771	1

```

        ! end of file indicates tape mark encountered
        KERNEL_CALL(ADJTM, .TM);
        RETURN 0;

    END;

    IF .STATUS<0,16> EQL SSS_ENDOFTAPE
    THEN
        RETURN 1;

    USER_STATUS[0] = .STATUS;
    USER_STATUS[1] = SSS_FCPSPACERR;
    ERR_EXIT();
    END;

RETURN 1;

END;
                                ! end of routine

```

Address	Instruction	Comment	Hex
00000000G	04 0004 00000	.ENTRY SPACE, Save R2	0696
	7E D4 00002	CLRL -(SP)	0738
0870	04 AC DD 00004	PUSHL NUMBER	
	26 DD 00007	PUSHL #38	
	0000V 30 00009	BSBW ISSUE_IO	
	0C C0 0000C	ADDL2 #12, SP	
	50 D0 0000F	MOVL R0, STATUS	
	52 E8 00012	BLBS STATUS, 3\$	0740
	52 B1 00015	CMPW STATUS, #2160	0744
	1E 12 0001A	BNEQ 2\$	
	01 D0 0001C	MOVL #1, TM	0747
	04 AC D5 0001F	TSTL NUMBER	0749
	03 18 00022	BGEQ 1\$	
	01 CE 00024	MNEGL #1, TM	0751
	50 DD 00027 1\$:	PUSHL TM	0755
	01 DD 00029	PUSHL #1	
	5E DD 0002B	PUSHL SP	
	0000V CF 9F 0002D	PUSHAB ADJTM	
	04 FB 00031	CALLS #4, @#SYSS\$CMKRNL	
	19 11 00038	BRB 4\$	0756
0878	8F 52 B1 0003A 2\$:	CMPW STATUS, #2168	0760
	0E 13 0003F	BEQL 3\$	
	52 D0 00041	MOVL STATUS, USER_STATUS	0764
	0898 8F 3C 00046	MOVZWL #2200, USER_STATUS+4	0765
	00 BF 0004D	CHMU #0	0766
	01 D0 0004F 3\$:	MOVL #1, R0	0769
	04 00052	RET	
	50 D4 00053 4\$:	CLRL R0	0771
	04 00055	RET	

; Routine Size: 86 bytes, Routine Base: \$CODES + 00C1

LOG10
V04-000

: 393

0772 1

N 5
16-Sep-1984 02:23:24
14-Sep-1984 12:46:42

VAX-11 Bliss-32 V4.C-742
DISK\$VMSMASTER:[MTAACP.SRC]LOG10.B32;1 Page 12 (5)

LOG1
V04-

: R

: 7


```

395 0773 1 GLOBAL ROUTINE WRITE_TM : NOVALUE L$WRITE_TM =
396 0774 1
397 0775 1 !++
398 0776 1
399 0777 1 FUNCTIONAL DESCRIPTION:
400 0778 1 This routine writes one tape mark.
401 0779 1
402 0780 1 CALLING SEQUENCE:
403 0781 1 WRITE_TM()
404 0782 1
405 0783 1 INPUT PARAMETERS:
406 0784 1 none
407 0785 1
408 0786 1 IMPLICIT INPUTS:
409 0787 1 IO_CHANNEL
410 0788 1
411 0789 1 OUTPUT PARAMETERS:
412 0790 1 none
413 0791 1
414 0792 1 IMPLICIT OUTPUTS:
415 0793 1 IO_STATUS, USER_STATUS
416 0794 1 Tape mark written, tm count incremented.
417 0795 1
418 0796 1 ROUTINE VALUE:
419 0797 1 none
420 0798 1
421 0799 1 SIDE EFFECTS:
422 0800 1 SSS_FCPWRITERR - write failure
423 0801 1
424 0802 1 !--
425 0803 1
426 0804 2 BEGIN
427 0805 2
428 0806 2 EXTERNAL REGISTER
429 0807 2 COMMON_REG;
430 0808 2
431 0809 2 LOCAL
432 0810 2 STATUS; ! io status
433 0811 2
434 0812 2 STATUS = ISSUE_IO(IO$_WRITEOF, 0, 0);
435 0813 2
436 0814 2 IF NOT .STATUS AND .STATUS<0,16> NEQ SSS_ENDOFTAPE
437 0815 2 THEN
438 0816 3 BEGIN
439 0817 3 USER_STATUS[0] = .STATUS;
440 0818 3 USER_STATUS[1] = SSS_FCPWRITERR;
441 0819 3 ERR_EXIT();
442 0820 3 END;
443 0821 2
444 0822 2 KERNEL_CALL(ADJTM, 1);
445 0823 1 END; ! end of routine
```

7E 7C 00000 WRITE_TM::

			28 DD 00002	CLRQ	-(SP)	: 0812
		0000V	30 00004	PUSHL	#40	
	5E		0C C0 00007	BSBW	ISSUE_10	
	15		50 E8 0000A	ADDL2	#12, SP	
0878	8F		50 B1 0000D	BLBS	STATUS, 1\$: 0814
			0E 13 00012	CMPW	STATUS, #2168	
			50 D0 00014	BEQL	1\$	
C000G	CF		8F 3C 00019	MOVL	STATUS, USER_STATUS	: 0817
0000G	CF	08A0	00 BF 00020	MOVZWL	#2208, USER_STATUS+4	: 0818
			01 DD 00022 1\$:	CHMU	#0	: 0819
			01 DD 00024	PUSHL	#1	: 0822
			5E DD 00026	PUSHL	SP	
		0000V	CF 9F 00028	PUSHAB	ADJTM	
00000000G	9F		04 FB 0002C	CALLS	#4, @#SYSS\$CMKRNL	
			05 00033	RSB		: 0823

; Routine Size: 52 bytes, Routine Base: \$CODE\$ + 0117

; 446 0824 1


```

448 0825 1 GLOBAL ROUTINE ADJTM (NUMBER) : COMMON_CALL NOVALUE =
449 0826 1
450 0827 1 !++
451 0828 1
452 0829 1 FUNCTIONAL DESCRIPTION:
453 0830 1 This routine adjusts the tm count by the given number.
454 0831 1
455 0832 1 CALLING SEQUENCE:
456 0833 1 ADJTM(ARG1), called in kernel mode
457 0834 1
458 0835 1 INPUT PARAMETERS:
459 0836 1 ARG1 - signed number to adjust count by
460 0837 1
461 0838 1 IMPLICIT INPUTS:
462 0839 1 CURRENT_VCB[VCB$B_TM]
463 0840 1
464 0841 1 OUTPUT PARAMETERS:
465 0842 1 none
466 0843 1
467 0844 1 IMPLICIT OUTPUTS:
468 0845 1 CURRENT_VCB[VCB$B_TM]
469 0846 1 CURRENT_VCB[VCB$L_ST_RECORD]
470 0847 1
471 0848 1 ROUTINE VALUE:
472 0849 1 none
473 0850 1
474 0851 1 SIDE EFFECTS:
475 0852 1 none
476 0853 1
477 0854 1 !--
478 0855 1
479 0856 2 BEGIN
480 0857 2
481 0858 2 EXTERNAL REGISTER
482 0859 2 COMMON_REG;
483 0860 2
484 0861 2 LOCAL
485 0862 2 TM; ! number of tm's
486 0863 2
487 0864 2 TM = .CURRENT_VCB[VCB$B_TM];
488 0865 2 TM = .TM + .NUMBER;
489 0866 2
490 0867 2 ! Now adjust number so it is a number between 0 and 2
491 0868 2 !
492 0869 2
493 0870 2 IF .TM GEQ 3
494 0871 2 THEN
495 0872 2 TM = .TM - 3;
496 0873 2
497 0874 2 IF .TM LSS 0
498 0875 2 THEN
499 0876 2 TM = .TM + 3;
500 0877 2
501 0878 2 CURRENT_VCB[VCB$B_TM] = .TM;
502 0879 2 CURRENT_VCB[VCB$L_ST_RECORD] = .CURRENT_UCB[UCB$L_RECORD];
503 0880 1 END; ! end of routine
```

PC	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

```
; Routine Size: 41 bytes,    Routine Base: $CODE$ + 014B
```

: 504 0881 1


```

506 0882 1 GLOBAL ROUTINE ISSUE_IO (FUNCTION, P1, P2) : L$ISSUE_IO =
507 0883 1
508 0884 1 ++
509 0885 1
510 0886 1 FUNCTIONAL DESCRIPTION:
511 0887 1 This routine issues the I/O and if the device is offline or
512 0888 1 the volume is invalid it repositions it.
513 0889 1
514 0890 1
515 0891 1 CALLING SEQUENCE:
516 0892 1 ISSUE_IO(FUNCTION,P1,P2)
517 0893 1
518 0894 1 INPUT PARAMETERS:
519 0895 1 ARG1 - function code
520 0896 1 ARG2 - P1 parameter
521 0897 1 ARG3 - P2 parameter
522 0898 1
523 0899 1 IMPLICIT INPUTS:
524 0900 1 none
525 0901 1
526 0902 1 OUTPUT PARAMETERS:
527 0903 1 none
528 0904 1
529 0905 1 IMPLICIT OUTPUTS:
530 0906 1 none
531 0907 1
532 0908 1 ROUTINE VALUE:
533 0909 1 I/O status
534 0910 1
535 0911 1 SIDE EFFECTS:
536 0912 1 none
537 0913 1
538 0914 1 --
539 0915 1
540 0916 2 BEGIN
541 0917 2
542 0918 2 EXTERNAL REGISTER
543 0919 2 COMMON_REG;
544 0920 2
545 0921 2 LOCAL
546 0922 2 CUR_RECORD;
547 0923 2
548 0924 2 ! save current position
549 0925 2 !
550 0926 2 CUR_RECORD = .CURRENT_UCB[UCB$L_RECORD];
551 0927 2
552 0928 2 WHILE 1
553 0929 2 DO
554 0930 3 BEGIN
555 0931 3
556 0932 4 BEGIN
557 0933 4 LOCAL
558 0934 4 STATUS;
559 0935 4
560 P 0936 4 STATUS = $QIOW(EFN = EFN, CHAN = .IO_CHANNEL,
561 P 0937 4 FUNC = .FUNCTION OR IOSM_CLSEREXCP,
562 0938 4 IOSB = IO_STATUS, P1 = .PT, P2 = .P2);
```

```

: 563      0939 4
: 564      0940 4
: 565      0941 4
: 566      0942 4
: 567      0943 4
: 568      0944 4
: 569      0945 4
: 570      0946 4
: 571      0947 4
: 572      0948 4
: 573      0949 4
: 574      0950 4
: 575      0951 4
: 576      0952 4
: 577      0953 1

      IF NOT .STATUS
      THEN
        IO_STATUS = .STATUS;
      END;

      IF .IO_STATUS<0,16> NEQ SS$_MEDOFL AND .IO_STATUS<0,16> NEQ SS$_VOLINV
      THEN
        RETURN .IO_STATUS;

      REPOSITION(.CUR_RECORD);
      END;

      RETURN 1;
      END;

                                ! end of routine ISSUE_IO
```

```

                                50      0000G CF D0 00000 ISSUE_IO::
                                00B0      C0 DD 00005      MOVL      CURRENT_UCB, R0      : 0926
7E      08      AE 00000200      8F C9 00009      PUSHL      176(R0)
                                7E 7C 00012 1$:      BISL3      #512, FUNCTION, -(SP)      : 0938
                                7E 7C 00014      CLRQ      -(SP)
                                24      AE DD 00016      PUSHL      P2
                                24      AE DD 00019      PUSHL      P1
                                7E 7C 0001C      CLRQ      -(SP)
                                0000G CF 9F 0001E      PUSHAB     IO_STATUS
                                24      AE DD 00022      PUSHL      36(SP)
                                0000G CF DD 00025      PUSHL      IO_CHANNEL
                                01      DD 00029      PUSHL      #1
00000000G 00      0C FB 0002B      CALLS      #12, SYSSQIOW
                                05      50 E8 00032      BLBS      STATUS, 2$
                                0000G CF 50 D0 00035      MOVL      STATUS, IO_STATUS
01A4      8F      0000G CF B1 0003A 2$:      CMPW      IO_STATUS, #420
                                10 13 00041      BEQL      3$
0254      8F      0000G CF B1 00043      CMPW      IO_STATUS, #596
                                07 13 0004A      BEQL      3$
                                50      0000G CF D0 0004C      MOVL      IO_STATUS, R0
                                0B 11 00051      BRB      4$
                                04      AE DD 00053 3$:      PUSHL      CUR_RECORD
                                0000V 30 00056      BSBW      REPOSITION
                                5E      04 C0 00059      ADDL2     #4, SP
                                B4 11 0005C      BRB      1$
                                5E      08 C0 0005E 4$:      ADDL2     #8, SP
                                05 00061      RSB
                                : 0940
                                : 0942
                                : 0945
                                :
                                : 0947
                                : 0949
                                :
                                : 0928
                                : 0953
                                :
```

; Routine Size: 98 bytes, Routine Base: \$CODE\$ + 0174

; 578 0954 1


```
580 0955 1 GLOBAL ROUTINE SPACE_TM (NUMBER) : COMMON_CALL NOVALUE =
581 0956 1
582 0957 1 ++
583 0958 1
584 0959 1 FUNCTIONAL DESCRIPTION:
585 0960 1 This routine spaces a given number of tm's in either direction.
586 0961 1
587 0962 1 CALLING SEQUENCE:
588 0963 1 SPACE_TM(NUMBER)
589 0964 1
590 0965 1 INPUT PARAMETERS:
591 0966 1 ARG1 - number of tm's to space
592 0967 1 (if negative, space backward. if positive, space forward.)
593 0968 1
594 0969 1 IMPLICIT INPUTS:
595 0970 1 IO_CHANNEL
596 0971 1
597 0972 1 OUTPUT PARAMETERS:
598 0973 1 none
599 0974 1
600 0975 1 IMPLICIT OUTPUTS:
601 0976 1 TM count incremented to reflect tape postioned beyond the tm specified
602 0977 1 IO_STATUS, USER_STATUS
603 0978 1
604 0979 1 ROUTINE VALUE:
605 0980 1 none
606 0981 1
607 0982 1 SIDE EFFECTS:
608 0983 1 $$$FCPSPACERR - space failure
609 0984 1
610 0985 1 --
611 0986 1
612 0987 2 BEGIN
613 0988 2
614 0989 2 EXTERNAL REGISTER
615 0990 2 COMMON_REG;
616 0991 2
617 0992 2 EXTERNAL ROUTINE
618 0993 2 BLOCK,
619 0994 2 SYS$QIO : ADDRESSING_MODE (ABSOLUTE);
620 0995 2
621 0996 2 LOCAL
622 0997 2 CUR_RECORD, ! current position of tape
623 0998 2 STATUS; ! io status
624 0999 2
625 1000 2 CUR_RECORD = .CURRENT_UCB[UCB$L_RECORD];
626 1001 2
627 1002 2 WHILE 1
628 1003 2 DO
629 1004 2 BEGIN
630 1005 2 BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_NO_TM] = .NUMBER;
631 1006 2 $QIO( CHAN = .IO_CHANNEL,
632 1007 2 FUNC = IOS_SKIPFILE OR IOSM_CLSEREXCP,
633 1008 2 IOSB = BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_STATUS],
634 1009 2 ASTADR = UNBLOCK_SPACE,
635 1010 2 ASTPRM = .CURRENT_VCB,
636 1011 2 P1 = .NUMBER );
```

```

637      1012      ! Block the process to wait for function to be completed
638      1013      !
639      1014      BLOCK($FIELDMASK(VCB$V_WAIREWIND));
640      1015
641      1016      STATUS = .BLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_STATUS];
642      1017
643      1018      IF .STATUS<0,16> NEQ SSS_MEDOFL AND .STATUS<0,16> NEQ SSS_VOLINV
644      1019      THEN
645      1020      EXITLOOP;
646      1021
647      1022      REPOSITION(.CUR_RECORD);
648      1023      END;
649      1024
650      1025      IF NOT .STATUS AND .STATUS<0,16> NEQ SSS_ENDOFTAPE
651      1026      THEN
652      1027      BEGIN
653      1028      USER_STATUS[0] = .STATUS;
654      1029      USER_STATUS[1] = SSS_FCPSPACERR;
655      1030      ERR_EXIT();
656      1031      END;
657      1032
658      1033      KERNEL_CALL(ADJTM, .NUMBER);
659      1034      END;
660      1035      ! end of routine
```

```

                                .EXTRN  BLOCK, SYSSQIO
                                .ENTRY   SPACE TM, Save R2,R3
                                MOV      CURRENT_UCB, R0
                                MOV      176(R0), CUR_RECORD
                                MOV      60(CURRENT_VCB), R0
                                MOV      NUMBER, 452(R0)
                                CLRQ     -(SP)
                                CLRQ     -(SP)
                                CLRL     -(SP)
                                PUSHL    NUMBER
                                PUSHL    CURRENT_VCB
                                PUSHAB   UNBLOCK_SPACE
                                PUSHAB   412(R0)
                                MOVZWL   #549, -(SP)
                                PUSHL    IO_CHANNEL
                                CLRL     -(SP)
                                CALLS    #12, SYSSQIO
                                PUSHL    #8
                                CALLS    #1, BLOCK
                                MOV      60(CURRENT_VCB), R0
                                MOV      412(R0), STATUS
                                CMPW     STATUS, #420
                                BEQL     2$
                                CMPW     STATUS, #596
                                BNEQ     3$
                                PUSHL    CUR_RECORD
                                BSBW     REPOSITION
                                ADDL2    #4, SP
                                BRB      1$
                                000C 00000
                                50 0000G CF D0 00002
                                53 00B0 CO D0 00007
                                50 3C AB D0 0000C 1$:
                                01C4 C0 04 AC D0 00010
                                7E 7C 00016
                                7E 7C 00018
                                7E D4 0001A
                                04 AC DD 0001C
                                5B DD 0001F
                                0000V CF 9F 00021
                                019C C0 9F 00025
                                7E 0225 8F 3C 00029
                                0000G CF DD 0002E
                                7E D4 00032
                                00000000G 00 0C FB 00034
                                08 DD 0003B
                                0000G CF 01 FB 0003D
                                50 3C AB D0 00042
                                52 019C C0 D0 00046
                                01A4 8F 52 B1 0004B
                                0254 8F 07 13 00050
                                52 B1 00052
                                0A 12 00057
                                53 DD 00059 2$:
                                0000V 30 0005B
                                5E 04 C0 0005E
                                A9 11 00061
                                .ENTRY   SPACE TM, Save R2,R3
                                MOV      CURRENT_UCB, R0
                                MOV      176(R0), CUR_RECORD
                                MOV      60(CURRENT_VCB), R0
                                MOV      412(R0), STATUS
                                CMPW     STATUS, #420
                                BEQL     2$
                                CMPW     STATUS, #596
                                BNEQ     3$
                                PUSHL    CUR_RECORD
                                BSBW     REPOSITION
                                ADDL2    #4, SP
                                BRB      1$
                                0955
                                1000
                                1005
                                1011
                                1015
                                1017
                                1019
                                1023
                                1002
```


LOGIO
V04-000

J 6
16-Sep-1984 02:23:24
14-Sep-1984 12:46:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]LOGIO.B32;1 Page 21
(9)

0878	15	52	E8	00063	3\$:	BLBS	STATUS, 4\$:	1026
	8F	52	B1	00066		CMPW	STATUS, #2168	:	
		0E	13	0006B		BEQL	4\$:	
0000G	CF	52	D0	0006D		MOVL	STATUS, USER_STATUS	:	1029
0000G	CF	8F	3C	00072		MOVZWL	#2200, USER_STATUS+4	:	1030
		00	BF	00079		CHMU	#0	:	1031
		04	AC	DD 0007B	4\$:	PUSHL	NUMBER	:	1034
		01	DD	0007E		PUSHL	#1	:	
		5E	DD	00080		PUSHL	SP	:	
		CF	9F	00082		PUSHAB	ADJTM	:	
00000000G	9F	04	FB	00086		CALLS	#4, @#SYSS\$CMKRNL	:	
		04	00	0008D		RET		:	1035

; Routine Size: 142 bytes, Routine Base: \$CODE\$ + 01D6

; 661 1036 1

```

663 1037 1 GLOBAL ROUTINE REPOSITION (NO_RECORD) : L$REPOSITION NOVALUE =
664 1038 1
665 1039 1 ++
666 1040 1
667 1041 1 FUNCTIONAL DESCRIPTION:
668 1042 1 This routine mounts the device that is offline and repositions
669 1043 1 to the current position.
670 1044 1
671 1045 1 CALLING SEQUENCE:
672 1046 1 REPOSITION(ARG1)
673 1047 1
674 1048 1 INPUT PARAMETERS:
675 1049 1 ARG1 - number of record to position to
676 1050 1
677 1051 1 IMPLICIT INPUTS:
678 1052 1 CURRENT_UCB - address of current unit control block
679 1053 1 CURRENT_VCB - address of current volume control block
680 1054 1
681 1055 1 OUTPUT PARAMETERS:
682 1056 1 none
683 1057 1
684 1058 1 IMPLICIT OUTPUTS:
685 1059 1 none
686 1060 1
687 1061 1 ROUTINE VALUE:
688 1062 1 none
689 1063 1
690 1064 1 SIDE EFFECTS:
691 1065 1 none
692 1066 1
693 1067 1 USER ERRORS:
694 1068 1 none
695 1069 1
696 1070 1 --
697 1071 1
698 1072 2 BEGIN
699 1073 2
700 1074 2 EXTERNAL REGISTER
701 1075 2 COMMON_REG;
702 1076 2
703 1077 2 LABEL
704 1078 2 OFFLINE;
705 1079 2
706 1080 2 LOCAL
707 1081 2 CVT_DEVNAM : VECTOR [MAX_DEVNAM_LENGTH,BYTE], ! Converted dev name
708 1082 2 CVT_DEVNAM_LENGTH : BYTE, ! and length of dev name
709 1083 2 SAV_TM,
710 1084 2 SAV_ST_REC,
711 1085 2 MVL_ENTRY, ! address of cur volume MVL entry
712 1086 2 VOL; ! current volume
713 1087 2
714 1088 2 VOL = .CURRENT_VCB[VCB$B_CUR_RVN];
715 1089 2 SAV_TM = .CURRENT_VCB[VCB$B_TM];
716 1090 2 SAV_ST_REC = .CURRENT_VCB[VCB$L_ST_RECORD];
717 1091 2
718 1092 2 ! This next call will use the UCB address to get the device's name and
719 1093 2 ! will fill in the fields with that name and the length of the name.
```



```

: 720      1094  2      GET_DEV_NAME(CVT_DEVNAM_LENGTH,CVT_DEVNAM);
: 721      1095  2
: 722      1096  2
: 723      1097  2      ! Set device not mounted since rewind does not currently recognize device
: 724      1098  2      ! offline
: 725      1099  2
: 726      1100  2      MVL_ENTRY = .CURRENT_VCB[VCB$L_MVL] + MVL$K_FIXLEN + ((.VOL - 1)*MVL$K_LENGTH);
: 727      1101  2
: 728      1102  2      OFFLINE :
: 729      1103  2      BEGIN
: 730      1104  2
: 731      1105  2      WHILE 1
: 732      1106  3      DO
: 733      1107  4          BEGIN
: 734      1108  4
: 735      1109  4          ! Send message to operator informing that the device is offline
: 736      1110  4          !
: 737      1111  4          PRINT_OPR_MSG(MOUN$ OFFLINE, 0, .CVT_DEVNAM_LENGTH,CVT_DEVNAM);
: 738      1112  4          KERNEC_CACL(RESET_UNIT);
: 739      1113  4
: 740      1114  4          ! Mount volume again
: 741      1115  4          MOUNT_VOL(.VOL,
: 742      1116  4              $FIELDMASK(MOUSV_REWIND) + $FIELDMASK(MOUSV_LBLCHECK) +
: 743      1117  4              $FIELDMASK(MOUSV_MOUNTERR));
: 744      1118  4
: 745      1119  4          WHILE 1
: 746      1120  4          DO
: 747      1121  5              BEGIN
: 748      1122  6              (
: 749      1123  6
: 750      1124  6              LOCAL
: 751      1125  6                  STATUS;
: 752      1126  6
: 753      1127  6              ! Space the number of blocks left to space
: 754      1128  6              !
: 755      1129  6              STATUS = $QIOW(EFN = EFN,
: 756      1130  6                  CHAN = .IO_CHANNEL,
: 757      1131  6                  FUNC = IOS_SKIPRECORD OR IOSM_CLSEREXCP,
: 758      1132  6                  IOSB = IO_STATUS,
: 759      1133  6                  P1 = .NO_RECORD - .CURRENT_UCB[UCB$L_RECORD]);
: 760      1134  6
: 761      1135  6              IF NOT .STATUS
: 762      1136  6              THEN
: 763      1137  5                  .IO_STATUS = .STATUS);          ! directive status
: 764      1138  5
: 765      1139  5              IF .NO_RECORD EQL .CURRENT_UCB[UCB$L_RECORD]
: 766      1140  5              THEN
: 767      1141  5                  LEAVE OFFLINE;          ! repositioning complete
: 768      1142  5
: 769      1143  5              IF .IO_STATUS<0,16> EQL SS$_MEDOFL OR .IO_STATUS<0,16> EQL SS$_VOLINV
: 770      1144  5                  OR
: 771      1145  5                  .CURRENT_UCB[UCB$L_RECORD] GEQ .NO_RECORD
: 772      1146  5              THEN
: 773      1147  5                  EXITLOOP          ! start again
: 774      1148  5              ELSE
: 775      1149  5
: 776      1150  5                  IF .IO_STATUS<0,16> NEQ SS$_ENDOFFILE

```

Address	Op Code	Register	Op Code	Register	Instruction	Address
5E	14	C2	00000		REPOSITION::	
7E	2F	AB	9A	00003	SUBL2	#20, SP
7E	2E	AB	9A	00007	MOVZBL	47(CURRENT_VCB), VOL
	30	AB	DD	0000B	MOVZBL	46(CURRENT_VCB), SAV_TM
	10	AE	9F	0000E	PUSHL	48(CURRENT_VCB)
	10	AE	9F	00011	PUSHAB	CVT_DEVNAM
0000G	CF	02	FB	00014	PUSHAB	CVT_DEVNAM_LENGTH
51	08	AE	D0	00019	CALLS	#2, GET_DEV_NAME
50	34	BB41	7E	0001D	MOVL	VOL, R1
50		1C	C0	00022	MOVAQ	@52(CURRENT_VCB)[R1], MVL_ENTRY
	10	AE	9F	00025	ADDL2	#28, MVL_ENTRY
7E	10	AE	9A	00028	PUSHAB	CVT_DEVNAM
	7E	D4	0002C		MOVZBL	CVT_DEVNAM_LENGTH, -(SP)
0072811C	8F	DD	0002E		CLRL	-(SP)
	0000G	30	00034		PUSHL	#7504156
5E	0C	C0	00037		BSBW	PRINT_OPR_MSG
	6E	D4	0003A		ADDL2	#12, SP
	5E	DD	0003C		CLRL	(SP)
0000G	9F	0000G	CF	9F	PUSHL	SP
	03	FB	00042		PUSHAB	RESET_UNIT
	0B	DD	00049		CALLS	#3, @#SYSS\$CMKRNL
0000G	0C	AE	DD	0004B	PUSHL	#11
	02	FB	0004E		PUSHL	VOL
	7E	7C	00053		CALLS	#2, MOUNT_VOL
	7E	7C	00055		CLRQ	-(SP)
	7E	D4	00057		CLRQ	-(SP)
38	50	0000G	CF	D0	CLRL	-(SP)
	AE	00B0	C0	C3	MOVL	CURRENT_UCB, R0
			7E	7C	SUBL3	176(R0), NO_RECORD, -(SP)
		0000G	CF	9F	CLRQ	-(SP)
	0226	8F	3C	0006B	PUSHAB	IO_STATUS
	0000G	CF	DD	00070	MOVZWL	#550, -(SP)
		01	DD	00074	PUSHL	IO_CHANNEL
0000G	00	0C	FB	00076	PUSHL	#1
	05	50	E8	0007D	CALLS	#12, SYSS\$QIOW
0000G	DF	50	D0	00080	BLBS	STATUS, 3\$
	50	0000G	CF	D0	MOVL	STATUS, @IO_STATUS
					MOVL	CURRENT_UCB, R0

LOGIO
V04-000

N 6
16-Sep-1984 02:23:24
14-Sep-1984 12:46:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]LOGIO.B32;1
Page 25
(10)

00B0	C0	24	AE	D1	0008A	CMP	NO_RECORD, 176(R0)	:	
			37	13	00090	BEQ	5\$:	
	51	0000G	CF	3C	00092	MOVZWL	IO_STATUS, R1	:	1143
01A4	8F		51	B1	00097	CMPW	R1, #420	:	
			87	13	0009C	BEQ	1\$:	
0254	8F		51	B1	0009E	CMPW	R1, #596	:	
			80	13	000A3	BEQ	1\$:	
24	AE	00B0	C0	D1	000A5	CMP	176(R0), NO_RECORD	:	1145
			03	19	000AB	BLSS	4\$:	
			FF75	31	000AD	BRW	1\$:	
0870	8F		51	B1	000B0	CMPW	R1, #2160	:	1150
			9C	13	000B5	BEQ	2\$:	
0000G	CF	0000G	CF	D0	000B7	MOVL	IO_STATUS, USER_STATUS	:	1153
0000G	CF	0988	8F	3C	000BE	MOVZWL	#240, USER_STATUS+4	:	1154
			00	BF	000C5	CHMU	#0	:	1155
			8A	11	000C7	BRB	2\$:	1119
			6E	DD	000C9	PUSHL	SAV_ST_REC	:	1163
		08	AE	DD	000CB	PUSHL	SAV_TM	:	
			02	DD	000CE	PUSHL	#2	:	
			5E	DD	000D0	PUSHL	SP	:	
		0000V	CF	9F	000D2	PUSHAB	RESTORE_POS	:	
00000000G	9F		05	FB	000D6	CALLS	#5, @SYS\$CMKRNL	:	
	5E		20	C0	000DD	ADDL2	#32, SP	:	1164
			05	000E0	RSB			:	

; Routine Size: 225 bytes, Routine Base: \$CODE\$ + 0264

; 791 1165 1

MAI
V04

```
: 793 1166 1 GLOBAL ROUTINE RESTORE_POS (TM, REC) : COMMON_CALL NOVALUE =
: 794 1167 1
: 795 1168 1 ++
: 796 1169 1
: 797 1170 1 FUNCTIONAL DESCRIPTION:
: 798 1171 1 This routine restores the tape position information destroyed by
: 799 1172 1 ASSUME_MOUNTED.
: 800 1173 1
: 801 1174 1 CALLING SEQUENCE:
: 802 1175 1 RESTORE_POS(ARG1,ARG2), in kernel mode
: 803 1176 1
: 804 1177 1 INPUT PARAMETERS:
: 805 1178 1 ARG1 - number or tape marks
: 806 1179 1 ARG2 - number of blocks into tape since last tape mark
: 807 1180 1
: 808 1181 1 IMPLICIT INPUTS:
: 809 1182 1 address of current VCB
: 810 1183 1
: 811 1184 1 OUTPUT PARAMETERS:
: 812 1185 1 none
: 813 1186 1
: 814 1187 1 IMPLICIT OUTPUTS:
: 815 1188 1 CURRENT_VCB[VCB$B_TM] and CURRENT_VCB[VCB$L_ST_RECORD] updated
: 816 1189 1
: 817 1190 1 ROUTINE VALUE:
: 818 1191 1 none
: 819 1192 1
: 820 1193 1 SIDE EFFECTS:
: 821 1194 1 none
: 822 1195 1
: 823 1196 1 --
: 824 1197 1
: 825 1198 2 BEGIN
: 826 1199 2
: 827 1200 2 EXTERNAL REGISTER
: 828 1201 2 COMMON_REG;
: 829 1202 2
: 830 1203 2 CURRENT_VCB[VCB$B_TM] = .TM;
: 831 1204 2 CURRENT_VCB[VCB$L_ST_RECORD] = .REC;
: 832 1205 1 END;
```

```
0000 00000
2E AB 04 AC 90 00002
30 AB 08 AC D0 00007
04 0000C
```

```
.ENTRY RESTORE_POS, Save nothing
MOVB TM, 46(CURRENT_VCB)
MOVL REC, 48(CURRENT_VCB)
RET
```

```
: 1166
: 1203
: 1204
: 1205
```

; Routine Size: 13 bytes, Routine Base: \$CODE\$ + 0345


```

834 1206 1 ROUTINE UNBLOCK_SPACE (VCB) : COMMON_CALL NOVALUE =
835 1207 1
836 1208 1 ++
837 1209 1
838 1210 1 FUNCTIONAL DESCRIPTION:
839 1211 1 This routine unblocks after a SPACE_TM has been done. If I/O
840 1212 1 is canceled, the tape position is updated.
841 1213 1
842 1214 1 calling sequence:
843 1215 1 UNBLOCK_SPACE()
844 1216 1
845 1217 1 INPUT PARAMETERS:
846 1218 1 ARG1 - address of volume control block
847 1219 1
848 1220 1 IMPLICIT INPUTS:
849 1221 1 VVPSL_NO_TM - number of tape marks positioned.
850 1222 1 ( If neg then backwards, else forwards.)
851 1223 1 Saved stack and impure area
852 1224 1
853 1225 1 OUTPUT PARAMETERS:
854 1226 1 none
855 1227 1
856 1228 1 IMPLICIT OUTPUTS:
857 1229 1 CURRENT_VCB[VCB$$_ST_RECORD]
858 1230 1
859 1231 1 ROUTINE VALUE:
860 1232 1 none
861 1233 1
862 1234 1 SIDE EFFECT:
863 1235 1 Never returns to PC where AST's were enabled.
864 1236 1 Instead it resumes where the blocked request left off.
865 1237 1
866 1238 1 --
867 1239 1
868 1240 2 BEGIN
869 1241 2
870 1242 2 EXTERNAL
871 1243 2 IO_PACKET;
872 1244 2
873 1245 2 LOCAL
874 1246 2 STATUS;
875 1247 2
876 1248 2 EXTERNAL ROUTINE
877 1249 2 DO_CANCEL : COMMON_CALL, ! cancel i/o
878 1250 2 UNBLOCK; ! unblock processing
879 1251 2
880 1252 2 EXTERNAL REGISTER
881 1253 2 COMMON_REG;
882 1254 2
883 1255 2 CURRENT_VCB = .VCB;
884 1256 2
885 1257 2 ! If cancel I/O request came thru while spacing tape mark, then restore
886 1258 2 UCB address, adjust number of tape marks if successful, and cancel
887 1259 2 request
888 1260 2
889 1261 2 IF .CURRENT_VCB[VCB$$_CANCELIO]
890 1262 2 THEN
```



```

: 891      1263      3      BEGIN
: 892      1264      3      CURRENT_UCB = (.CURRENT_VCB[VCB$L_VPBL] + VVP$K_LENGTH + (CURRENT_UCB
: 893      1265      3      - USER_STATUS[0]));
: 894      1266      3
: 895      1267      3      IF .STATUS<0,16> EQL SSS_ENDOFTAPE OR .STATUS
: 896      1268      3      THEN
: 897      1269      3          KERNEL_CALL(ADJTM, .BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_NO_TM]);
: 898      1270      3
: 899      1271      3      ERROR(SS$ CANCEL);
: 900      1272      3      KERNEL_CALL(DO_CANCEL);
: 901      1273      3      IO_PACKET = 0;
: 902      1274      3      RETURN;
: 903      1275      3
: 904      1276      3      END;
: 905      1277      3
: 906      1278      3      ! Unblock process and continue where request processing left off.
: 907      1279      3      !
: 908      1280      3      UNBLOCK();
: 909      1281      3      END;
: INFO#250      L1:1267      ! end of routine
: Referenced LOCAL symbol STATUS is probably not initialized
```

				.EXTRN	IO_PACKET, DO_CANCEL						
				.EXTRN	UNBLOCK						
0004 00000 UNBLOCK_SPACE:											
		52	00000000G	9F	9E	00002	.WORD	Save R2		1206	
		5B	04	AC	D0	00009	MOVAB	@#SY\$CMKRNL, R2			
42	0B	AB		05	E1	0000D	MOVL	VCB, CURRENT_VCB		1255	
50	40	AB	00000000*	8F	C1	00012	BBC	#5, 11(CURRENT_VCB), 3\$		1261	
							ADDL3	#<<CURRENT_UCB=USER_STATUS>+12>, -		1264	
								64(CURRENT_VCB), R0			
	0000G	CF		60	D0	0001B	MOVL	(R0), CURRENT_UCB			
	0878	8F		50	B1	00020	CMPL	STATUS, #2168		1267	
				03	13	00025	BEQL	1\$			
		13		50	E9	00027	BLBC	STATUS, 2\$			
		50					MOVL	60(CURRENT_VCB), R0		1269	
			3C	AB	D0	0002A	1\$:				
			01C4	C0	DD	0002E	PUSHL	452(R0)			
				01	DD	00032	PUSHL	#1			
				5E	DD	00034	PUSHL	SP			
			FDBF	CF	9F	00036	PUSHAB	ADJTM			
		62		04	FB	0003A	CALLS	#4, SY\$CMKRNL			
	0000G	CF	0830	8F	B0	0003D	2\$:	MOVW	#2096, USER_STATUS		1271
				7E	D4	00044	CLRL	-(SP)		1272	
				5E	DD	00046	PUSHL	SP			
			0000G	CF	9F	00048	PUSHAB	DO_CANCEL			
		62		03	FB	0004C	CALLS	#3, SY\$CMKRNL			
			0000G	CF	D4	0004F	CLRL	IO_PACKET		1273	
					04	00053	RET			1263	
	0000G	CF		00	FB	00054	3\$:	CALLS	#0, UNBLOCK		1280
					04	00059	RET			1281	

; Routine Size: 90 bytes, Routine Base: \$CODE\$ + 0352


```

911 1282 1 GLOBAL ROUTINE CHCK_IO_CLR_EXCP : COMMON_CALL NOVALUE =
912 1283 1
913 1284 1 ++
914 1285 1
915 1286 1 FUNCTIONAL DESCRIPTION:
916 1287 1 This routine saves the drives characteristics than does
917 1288 1 a QIOW set mode to the device to ensure that
918 1289 1 all outstanding reads or writes have been posted to the VCB
919 1290 1 before processing continues. This is necessary to ensure consistant
920 1291 1 behaviour between the old class of tape drives and the new type
921 1292 1 which speak tape protocol. The old tape drivers will still
922 1293 1 put all outstanding IO's on the VCB's blocked IO queue. The new
923 1294 1 drivers will complete these IO's with an error of SSS_SERIOUSEXCP
924 1295 1 which the ACP will queue on it's blocked IO queue.
925 1296 1
926 1297 1 CALLING SEQUENCE:
927 1298 1 CHCK_IO_CLR_EXCP()
928 1299 1
929 1300 1 INPUT PARAMETERS:
930 1301 1 none
931 1302 1
932 1303 1 IMPLICIT INPUTS:
933 1304 1 IO CHANNEL
934 1305 1 CURRENT_UCB
935 1306 1
936 1307 1 OUTPUT PARAMETERS:
937 1308 1 none
938 1309 1
939 1310 1 IMPLICIT OUTPUTS:
940 1311 1 none
941 1312 1
942 1313 1 ROUTINE VALUE:
943 1314 1 none
944 1315 1
945 1316 1 --
946 1317 1
947 1318 2 BEGIN
948 1319 2
949 1320 2 EXTERNAL REGISTER
950 1321 2 COMMON_REG;
951 1322 2
952 1323 2 LOCAL
953 1324 2 SAVE_DEVCHAR : VECTOR [2], ! Characteristics of drive
954 1325 2 STATUS; ! io status
955 1326 2
956 1327 2 SAVE_DEVCHAR [0] = .(CURRENT_UCB[UCB$B_DEVCLASS])<0,32>;
957 1328 2 SAVE_DEVCHAR [1] = .CURRENT_UCB[UCB$L_DEVDEPEND];
958 1329 2 STATUS = ISSUE_IO ( IO$_SETMODE, SAVE_DEVCHAR, 0);
959 1330 2
960 1331 1 END;
```

```

5E          0000 00000          .ENTRY CHCK_IO_CLR_EXCP, Save nothing          : 1282
          04 C2 00002          SUBL2 #4, SP          :
```

LOGIO
V04-000

F 7
16-Sep-1984 02:23:24
14-Sep-1984 12:46:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]LOGIO.B32;1 Page 30 (13)

50	0000G	CF	D0	00005	MOVL	CURRENT_UCB, R0	:	1327	
	40	A0	DD	0000A	PUSHL	64(R0)	:		
04	AE	44	A0	D0	0000D	MOVL	68(R0), SAVE_DEVCHAR+4	:	1328
		7E	D4	00012	CLRL	-(SP)	:	1329	
	04	AE	9F	00014	PUSHAB	SAVE_DEVCHAR	:		
		23	DD	00017	PUSHL	#35	:		
		FDAC	30	00019	BSBW	ISSUE_IO	:		
			04	0001C	RET		:	1331	

; Routine Size: 29 bytes, Routine Base: \$CODE\$ + 03AC

; 961 1332 1


```

: 963      1333 1 GLOBAL ROUTINE COMPLETE_VIO : COMMON_CALL NOVALUE =
: 964      1334 1
: 965      1335 1 ++
: 966      1336 1
: 967      1337 1 FUNCTIONAL DESCRIPTION:
: 968      1338 1     This routine gets all I/O's queued off the VCB's blocked queue and
: 969      1339 1     completes them to the user with an ABORT status
: 970      1340 1
: 971      1341 1 CALLING SEQUENCE:
: 972      1342 1     KERNEL_CALL(COMplete_VIO)
: 973      1343 1
: 974      1344 1 INPUT PARAMETERS:
: 975      1345 1     none
: 976      1346 1
: 977      1347 1 IMPLICIT INPUTS:
: 978      1348 1     none
: 979      1349 1
: 980      1350 1 OUTPUT PARAMETERS:
: 981      1351 1     none
: 982      1352 1
: 983      1353 1 IMPLICIT OUTPUTS:
: 984      1354 1     none
: 985      1355 1
: 986      1356 1 ROUTINE VALUE:
: 987      1357 1     none
: 988      1358 1
: 989      1359 1 SIDE EFFECTS:
: 990      1360 1     All outstanding IO's will be completed in error to the user.
: 991      1361 1
: 992      1362 1 --
: 993      1363 1
: 994      1364 2 BEGIN
: 995      1365 2
: 996      1366 2 EXTERNAL REGISTER
: 997      1367 2     COMMON_REG;
: 998      1368 2
: 999      1369 2 LOCAL
1000      1370 2     PACKET : REF BBLOCK;                ! address of io request packet
1001      1371 2
1002      1372 2 WHILE 1
1003      1373 2 DO
1004      1374 2     BEGIN
1005      1375 2         IF REMQUE (.CURRENT_VCB[VCB$L_BLOCKFL], PACKET)
1006      1376 2         THEN EXITLOOP;
1007      1377 2
1008      1378 2 ! make the error an ABORT status
1009      1379 2
1010      1380 2     PACKET[IRP$L_IOST1] = SSS ABORT;
1011      1381 2     USER_STATUS[0] = .PACKET[IRP$L_IOST1];
1012      1382 2     USER_STATUS[1] = .PACKET[IRP$L_IOST2];
1013      1383 2     KERNEL_CALL(IO_DONE, .PACKET);
1014      1384 2     END
: 1015      1385 1 END;
```

	52	00	BB	0F	00002	1\$:	.ENTRY	COMPLETE VIO, Save R2	:	1333
			1D	1D	00006		REMQUE	@0(CURRENT_VCB), PACKET	:	1375
	38	A2	2C	D0	00008		BVS	2\$:	
0000G	CF	38	A2	7D	0000C		MOVL	#44, 56(PACKET)	:	1380
			52	DD	00012		MOVQ	56(PACKET), USER_STATUS	:	1381
			01	DD	00014		PUSHL	PACKET	:	1383
			5E	DD	00016		PUSHL	#1	:	
		0000G	CF	9F	00018		PUSHL	SP	:	
00000000G	9F		04	FB	0001C		PUSHAB	IO_DONE	:	
			DD	11	00023		CALLS	#4, @#SYSS\$CMKRN	:	
			04	00	00025	2\$:	BRB	1\$:	1372
							RET		:	1385

; Routine Size: 38 bytes, Routine Base: \$CODE\$ + 03C9

; 1016 1386 1 END
; 1017 1387 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	1007	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	45	0	1000	00:01.9

; Information: 1
; Warnings: 0
; Errors: 0

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:LOG10/OBJ=OBJ\$:LOG10 MSRC\$:LOG10/UPDATE=(ENHS\$:LOG10)

; Size: 1007 code + 0 data bytes
; Run Time: 00:22.4
; Elapsed Time: 01:05.6

LOGIO
V04-000

¹₇
16-Sep-1984 02:23:24

VAX-11 Bliss-32 V4.0-742

Page 33

: Lines/CPU Min: 3720
: Lexemes/CPU-Min: 19424
: Memory Used: 126 pages
: Compilation Complete

MA1
V04

0255 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

